

REMARKS

Claims 1-71 are pending in the present application. Claims 1, 41, 45, 49, 58, and 65 have been amended herein. Claims 13, 38, 39, 40, 57, 59, 60, 61, 62, 63, 64, 66, 67, 68, 69, 70, and 71 have been cancelled. No new matter has been added.

The Examiner rejected claims 1-10, 38-40, 49-52, 55-56, 58-59, and 65-66 under 35 U.S.C § 102(e) as being anticipated by U.S. Patent No. 6,226,689 ("Shah"). Applicants respectfully submit that claim 1-10, 38-40, 49-52, 55-56, 58-59, and 65-66 are patentable over the art of record for the following reasons.

Independent claim 1 as amended recites:

A method in a computer system for servicing requests from one or more client computers, the method comprising:
receiving a request from a client computer;
a first thread processing the request by invoking a receive handler that creates a work item, wherein the first thread is part of a pool of generic threads;
a second thread performing a task specified in the work item by invoking a work handler, wherein the second thread is part of the pool of generic threads;
receiving a result of performing the task; ~~and~~
a third thread returning at least a portion of the result to the client computer by invoking a reply handler, wherein the third thread is part of the pool of generic threads.
the first thread, the second thread or the third thread indicating that the first thread, the second thread or the third thread has completed a work item; ~~and~~
if a quantum has not expired for the first thread, the second thread or the third thread, then the first thread, the second thread or the third thread being given an additional work item to perform without relinquishing the central processing unit upon which the first thread, the second thread or the third thread was running.

Claim 1 was amended to incorporate the features of now cancelled claim 13, that **if a quantum has not expired for the first thread, the second thread or the third thread, then the first thread, the second thread or the third thread being given an additional work item to perform without relinquishing the central processing unit upon which the first thread, the second thread or the third thread was running.**

Shah teaches a method for interprocess communication between a thread of a client application and a thread of a server application (Shah, Abstract). A client thread sends a request to a server listening thread, and the server listening thread places the request in a message queue associated with the server thread (Id.). The request is received at the server thread and dispatched to a remote procedure for processing (Id.). Reply data received back from the remote procedure is sent to a client listening thread, and the client listening thread notifies the client thread when the reply is received and gives the reply to the client thread (Id.).

Ramakrishnan teaches a scheduling method and apparatus for use in stations of a communications network (Ramakrishnan, col. 4, ll. 9-12). Processing tasks are divided into processing threads, each of which is structured to execute for a limited time before being subject to preemption by another processing thread (Id., col. 4, ll. 17-20). Each processing thread is given a weight that determines how long that processing thread is allowed to run before giving priority to another thread (Id., col. 4, ll. 33-35).

In the Office Action, the Examiner rejected claim 13 as being unpatentable over Shah in view of Ramakrishnan. Specifically, the Examiner stated that Ramakrishna taught the above feature at column 9, line 52 through column 10, line 3. To the contrary, the cited portion of Ramakrishnan teaches a general thread being allowed to run for a minimum execution time, after which priority is given to a real time processing thread (Ramakrishna, col. 9, ll. 59-61). There is no mention at all in the cited portion of determining if a quantum has expired, and if not, providing more work to a thread anywhere in the cited portion. It is therefore respectfully requested that the examiner withdraw the rejection and allow claim 1.

Claims 2-10 are all variously dependent on claim 1, and are therefore allowable for at least the reasons given for claim 1. It is therefore respectfully requested the Examiner withdraw the rejection and allow claims 2-10.

Claim 49, as amended, teaches:

An application program for implementation by an application server, the application program comprising:
at least one receive handler that can be invoked by a thread within a pool of threads, wherein each thread in the pool of threads is identical;
at least one work handler that also can be invoked by the thread;
at least one reply handler that also can be invoked by the thread;
and

one or more complex logic handlers that can be invoked by a second type of thread, wherein a thread of the second type of thread is executed when a request from a client computer involves execution of complex or long-running logic

Claim 49 was amended to include **one or more complex logic handlers that can be invoked by a second type of thread, wherein a thread of the second type of thread is executed when a request from a client computer involves execution of complex or long-running logic**. This feature was originally found in now cancelled claim 57. The Examiner originally rejected claim 57 under 35 U.S.C. 103(a) as being unpatentable over Shah in view of Challenger (Office Action, page 24).

Challenger teaches systems and methods for determining how changes in underlying data effect the value of objects (Challenger, col. 3, ll. 8-10). The methods include a method for specifying dependencies between objects and underlying data which allows a computer system to propagate updates to all objects in the system after an underlying data change (Id., col. 3, ll. 23-27). Another method manages relational objects whereby implicit data dependencies between the relational objects are automatically added by the object manager (Id., col. 3, ll. 60-62).

The Examiner stated that Challenger taught **one or more complex logic handlers that can be invoked by a second type of thread, wherein a thread of the second type of thread is executed when a request from a client computer involves execution of complex or long-running logic** at column 30, line 34 through column 33 lines 50 (Id.). Applicants respectfully submit that the cited portion of Challenger does not teach or suggest such features. At most the cited portion teaches the use of a trigger monitor, which is itself a long running process (Challenger, col. 30, lines 39-41). However, there is no mention of **one or more complex logic handlers that can be invoked by a second type of thread, wherein a thread of the second type of thread is executed when a request from a client computer involves execution of complex or long-running logic**, as required by the claims. It is therefore respectfully requested that the Examiner withdraw the rejection and allow claim 49.

Independent claims 58 and 65 as amended contains similar features as independent claim 49, and are therefore allowable for at least the reasons given above with respect to

claim 49. It is therefore respectfully requested that the Examiner withdraw the rejection and allow claims 58, and 65.

Claims 50-52, and 55-56 are variously dependent on independent claim 49, and are allowable for at least the reasons given for claim 49. It is therefore respectfully requested that the Examiner withdraw the rejection and allow claims 50-52, and 55-56.

Claims 26-32 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Shah in view of Fant (U.S. Patent No. 6,327,607) in view of Wight (U.S. Patent No. 6,219,353).

Claim 26 teaches:

A method in a computer system for servicing requests from multiple client computers, the method comprising:

determining that work is available after receiving a request from a client computer, wherein the request from the client computer is a request to perform a function having multiple states;

when work is available, a first work handler invoked by a first thread looking in a first work queue for a first work item corresponding to the work, wherein the first thread is a generic thread within a pool of generic threads; and

if the first work item is not found in the first work queue, the first work handler looking in a second work queue for the first work item.

Fant teaches an invocation architecture for generally concurrent process resolution comprising a plurality of interconnected processors, some of the processors being homogeneous processors and others being special purpose processors (Fant, col. 1, ll. 50-55). Each homogeneous processor is capable of invoking a connected processor to have the connected processor resolve processes (Id., col. 1, ll. 55-57). Each processor is capable of being invoked by a connected processor to resolve processes. (Id., col. 1, ll. 57-60).

Wight teaches a message communication system which performs efficient scheduling of access to a data communication medium by a plurality of nodes connected to the medium using simple signals and circuitry (Wight, col. 1, ll. 60-65). The system features a message hub that operates in a scheduling phase and a transmission phase (Id., col. 1, l. 65-col. 2, l. 1). The system schedules the access of each node to a data communication medium without requiring any special scheduling signal (Id., col. 2, ll. 1-3).

The Examiner admits that Shah does not teach **if the first work item is not found in the first work queue, the first work handler looking in a second work queue for the first**

work item, as required by claim 26 (Office Action, page 20). However, the Examiner states that Wight teaches such a feature at column 4, line 60 through line 3 (Id.). To the contrary, the cited portion of Wight teaches a message hub system, where messages are sent between nodes and processed according to the priority of their queue (Wight, column 4, ll 43-50). There is no mention of a work handler, or work item anywhere in the cited portion. In addition, even if Wight taught such a feature, which it doesn't, the Examiner has offered no motivation to combine the features of Wight with the teaching of Shah. It is therefore respectfully requested that the Examiner withdraw the rejection and allow claim 26.

Claims 27-32 are all variously dependant on independent claim 26, and are therefore allowable for at least the reasons given for the independent claim. It is therefore respectfully requested that the Examiner withdraw the rejection and allow claims 27-32.

Claims 41-48 are rejected under 35 U.S.C. § 102(b) as being anticipated by Bank (U.S. Patent No. 5,901,334).

Independent claim 41 as amended teaches

A method in a computer system for servicing requests from multiple client computers, the method comprising:
monitoring a quantity of work being performed by the computer system;
determining whether the quantity has exceeded an upper limit; and
if the quantity has exceeded the upper limit but has not dropped below a lower limit, not accepting new requests into the computer system,
wherein the upper and lower limits are not equal.

Banks teaches a method for the management of requests for a communication link between local and remote systems (Banks, abstract). An expected time an item will spend on a queue is calculated when an item is received (Id.). If this time exceeds a predetermined upper limit, then the item is rejected and the queue is purged of all remaining items (Id.). An indicator is provided to the sources of the purged items that they were removed from the queue (Id.).

Independent claim 41 was amended to clarify that the upper and lower limits are not equal. Nowhere does Banks teach monitoring a quantity of work being performed by the computer system, determining whether the quantity has exceeded an upper limit, and if the quantity has exceeded the upper limit but has not dropped below a lower limit, not accepting

new requests into the computer system, wherein the upper and lower limits are not equal. It is therefore respectfully requested that the Examiner withdraw the rejection and allow claim 41.

Independent claim 45 as amended, contains similar features as claim 41, and is therefore allowable for at least the reasons given for claim 41. It is therefore respectfully requested that the Examiner withdraw the rejection and allow claim 45.

Claims 42-44, and 46-48, are all variously dependant on independent claims 41 and 45, and are therefore allowable for at least the reasons given for the independent claims. It is therefore respectfully requested that the Examiner withdraw the rejection and allow the claims.

Claims 11-12, 14-16, 18-21, 23-25, 53-54, 60, and 67 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Shah in view of Fant.

Claims 11-12 are both dependent on independent claim 1, and are therefore allowable for at least the reasons given for claim 1. It is therefore respectfully requested that the Examiner withdraw the rejection and allow claims 11 and 12.

Claim 14 teaches:

A method in a computer system for servicing requests from multiple client computers, the method comprising:

receiving a request from a client computer to perform a multi-state function;

performing a first task, by a first work handler invoked by a first thread in a ready state, wherein the first task is associated with a first state of the multi-state function, and performing the first task includes issuing an asynchronous request for data;

placing the first thread back in the ready state;

receiving the data specified in the asynchronous request; and

performing a second task, by a second work handler invoked by a second thread in the ready state, wherein **the second task is associated with a second state of the multi-state function, and the second task performs an operation on the data**, wherein the first thread and the second thread are all identical generic threads within a pool of generic threads.

The Examiner states that Shah teaches **a second task associated with a multistate function performing an operation on the data specified in the asynchronous request**, at page 4, lines 58-67 (Office Action, page 13). To the contrary, the portion of Shah cited by the Examiner teaches a listening thread passing messages from a message queue to a window (Shah, col. 4, lines 58-67). There is no mention of a second state, or even a multistate

function anywhere in the cited portion. It is therefore respectfully requested that the Examiner withdraw the rejection and allow claim 14.

Claims 15, 16, 18-21, and 23-25 are all variously dependant on claim 14, and are therefore allowable for at least the reasons given for independent claim 14. It is therefore respectfully requested that the Examiner withdraw the rejection and allow claims 15, 16, 18-21, and 23-25.

Claims 53-54, and 60 are all variously dependent on independent claims 49, and 58, and are therefore allowable for at least the reasons given above for those claims. It is therefore respectfully requested that the Examiner withdraw the rejection and allow the claims.

Claims 17 and 22 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Shah in view of Fant in view of Wight.

Claims 17, and 22 are both variously dependant on claim 14, and are allowable for at least the reasons given for Shah and Fant. Wight fails to cure the deficiencies of Shah and Fant. It is therefore respectfully requested that the Examiner withdraw the rejection and allow claims 17, and 22.

Claims 33, 36, 57, 62, and 69 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Shah in view of Challenger (U.S. Patent No. 6,026,413).

Claim 33 teaches:

A method in a computer system for servicing requests from multiple client computers, the method comprising:
receiving, from a client computer, a request to perform a first task;
evaluating the first task, by a first handler invoked by a first thread, to determine whether the first task includes complex or long-running logic; and
if the first task includes complex or long-running logic, performing the first task by a second handler invoked by a second thread, wherein the first and second threads are all identical threads within a pool of generic threads.

The Examiner stated that Challenger taught **evaluating the first task, by a first handler invoked by a first thread, to determine whether the first task includes complex or long-running logic** at column 30, line 34 through column 33 lines 50 (Office Action, page 23). Applicants respectfully submit that the cited portion of Challenger does not teach or

DOCKET NO.: MSFT-0562
Application No.: 09/717,675
Office Action Dated: May 16, 2005

PATENT
REPLY FILED UNDER EXPEDITED
PROCEDURE PURSUANT TO
37 CFR § 1.116

suggest such features. At most the cited portion teaches the use of a trigger monitor, which is itself a long running process (Challenger, col. 30, lines 39-41). However, there is no mention of **evaluating the first task, by a first handler invoked by a first thread, to determine whether the first task includes complex or long-running logic**, as required by the claims. It is therefore respectfully requested that the Examiner withdraw the rejection and allow claim 33.

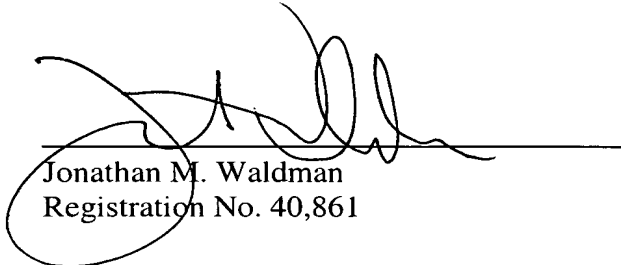
Claim 36 is dependent on claim 33, and is allowable for at least the reasons given above for claim 33. It is therefore respectfully requested that the Examiner withdraw the rejection and allow the claim.

Claims 34-35 and 37 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Shah in view of Challenger in view of Wight. Claims 34, 35, and 37, are all variously dependent on claim 33, and are therefore allowable for at least the reasons given above with respect to Challenger and Shah. Wight fails to cure the deficiencies of Challenger and Shah. It is therefore respectfully requested that the Examiner withdraw the rejection and allow claims 34, 35, and 37.

CONCLUSION

In view of the above remarks, Applicants respectfully submit that the present application is in condition for allowance. Reconsideration of the application and an early Notice of Allowance are respectfully requested.

Date: July 18, 2005



Jonathan M. Waldman
Registration No. 40,861

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439